

Ссылки, якоря <a> и фреймы

```
<a href="адрес">Ссылка</a>
```

В привычном для нас смысле ссылка в HTML оформляется с помощью элемента `<a>`. Это сокращение от английского *anchor*, что переводится как "якорь". А вот привычное слово link (ссылка) и соответствующий элемент `<link>` выполняет немного иную функцию, о которой мы скажем позднее.

Элемент `<a>` является **парным**. Его содержанием становится то, что мы хотим сделать ссылкой. Это может быть всё, что угодно: текст, изображения, таблицы, формы ввода, кнопки, блоки и т.п.

Атрибуты

У данного элемента есть обязательный атрибут - `href` (*hypertext reference* (слово *reference* также может быть переведено как ссылка или отсылка)), без которого ссылка вообще не выполняет свою функцию. Значением этого атрибута и становится адрес... (см. следующий шаг).

`target` - определяет, где будет открываться сайт. По умолчанию она открывается в текущей вкладке браузера. Установив этому атрибуту значение `_blank` ссылка будет открываться в новой вкладке.

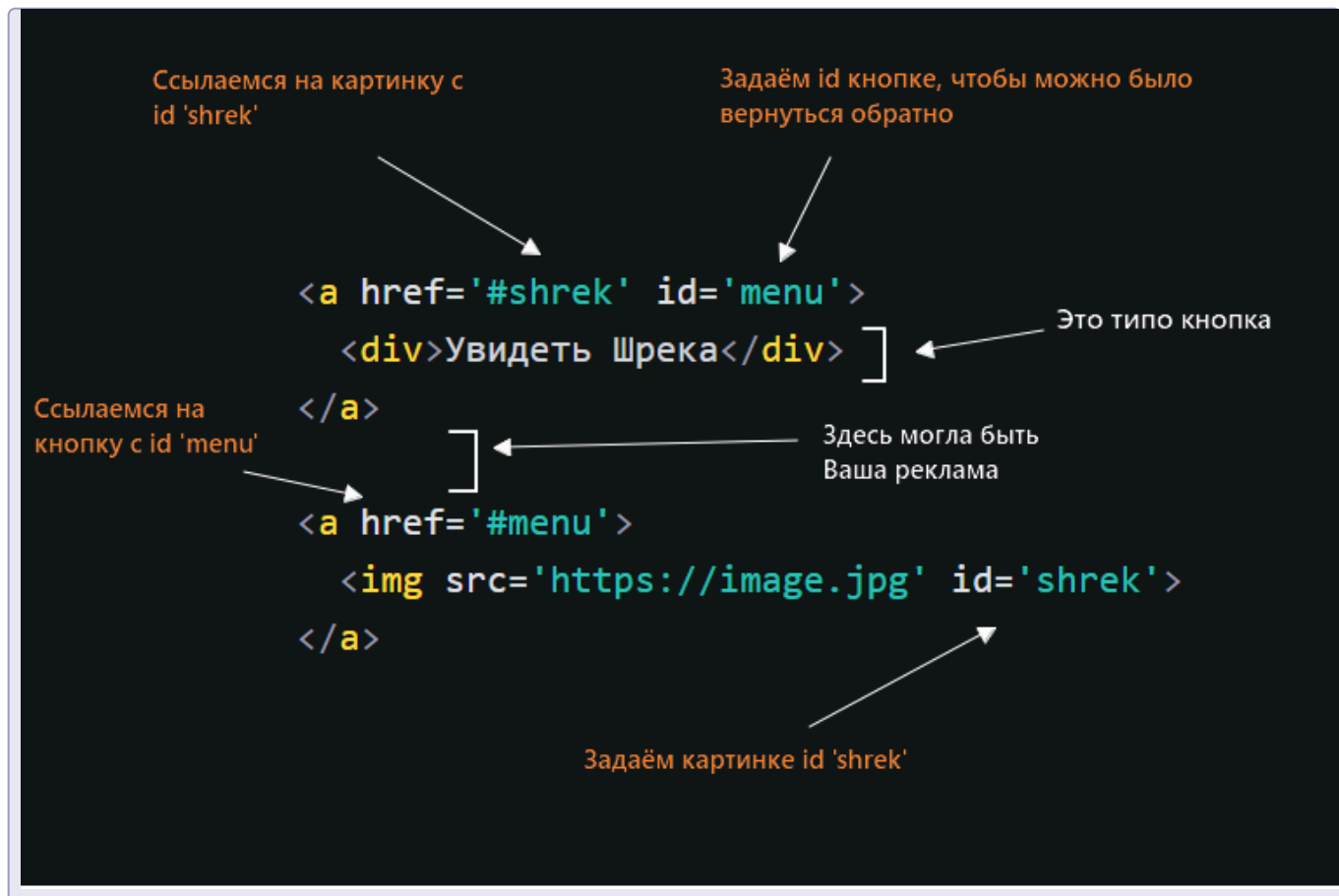
Якорь (метка)

Использование `<a>` как якоря (или метки) - неотъемлемая деталь лендингов. Якорь нужен для того, чтобы мы могли передвигаться по одной странице, к примеру, в шапке меню у нас есть кнопка, по нажатию на которую браузер перелистывает страницу в конец.

Для якоря необходимо:

- использовать универсальный атрибут `id`, в который мы должны прописать название нашей метки;

- из элемента `<a>` обратиться к метке через атрибут `href`, но перед именем метки установить символ решётки "`#`".



Т.е., чтобы ссылка переносила нас на конкретное место в документе, нужно задать элементу `id`, и сослаться на этот `id` в элементе `<a>`.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Minima quisquam dicta, veritatis ex inventore beatae hic pariatur vitae eius nostrum enim molestias, excepturi perspiciatis exercitationem eum quidem dolores provident facere sapiente corporis. Quae eaque illo debitis maiores quibusdam nihil unde doloremque vitae beatae expedita ducimus, porro voluptates deserunt alias a accusantium nobis? Vitae molestias natus dignissimos? Libero corporis totam tenetur, repudiandae iste consectetur mollitia numquam molestias ex fuga voluptate, blanditiis illo dignissimos ipsa alias cupiditate perferendis dolore. Rerum sunt deleniti aperiam quibusdam iste vitae maiores eveniet aut quo quam optio quaerat deserunt doloribus quos ullam omnis culpa minus delectus dolor voluptatibus nihil, doloremque aliquam esse? Consequuntur repellendus fuga, suscipit voluptatibus facere quis consequatur beatae consectetur ad nulla deserunt tempore eligendi inventore ullam quidem id blanditiis. Nisi omnis quisquam illum esse libero ullam, est, officiis, reiciendis quam minima autem dolor perspiciatis mollitia veniam! Omnis nisi natus doloremque quisquam vel praesentium vitae quaerat, magnam odit, cupiditate et tenetur quas nobis numquam molestias aperiam perferendis maiores fugiat facilis. Quaerat deleniti recusandae, illo suscipit obcaecati totam nesciunt, minus voluptates beatae magni itaque ratione magnam officiis. Asperiores suscipit eaque exercitationem nemo distinctio, quasi voluptatum et officiis hic rem cum. Fugit dicta eaque itaque officiis delectus?

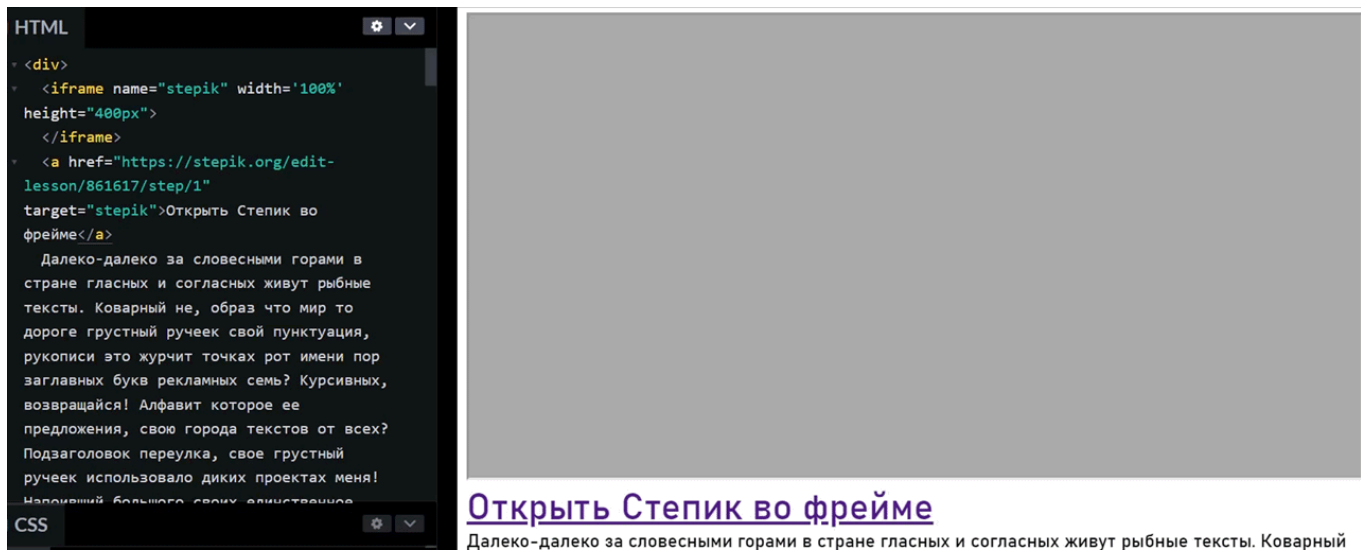
Использование вместе с <iframe>

В данном курсе я не буду подробно описывать элемент **iframe** (*inline frame*), только особенность его связи с элементом **<a>**.

<iframe> позволяет открывать на вашей странице другую страницу. К примеру, этот элемент используется для встраивания видео из YouTube или музыки из Яндекса. Это строчный парный элемент и его размеры определяют размер окошка, в котором может отображаться страница другого сайта или её часть (встраивание окошек CodePen в данный курс сделано тоже с помощью **<iframe>**).

Основой связи служит атрибут **target** элемента **<a>**. Дело в том, что в качестве его значения можно указать имя фрейма, которое задаётся фрейму атрибутом **name**. И тогда ссылка будет открываться во фрейме. Если атрибуту **target** установлено *несуществующее имя*, то будет открыта новая вкладка браузера.

Пример:



Помимо этого у фрейма есть атрибут `src`, в качестве значения которого можно указать URL, которая будет загружаться сразу с загрузкой страницы. Менять ширину и высоту фрейма можно с помощью атрибутов (о них подробнее рассказано в шаге про)

P.S.

Если сейчас вы не поняли тему меток, то не страшно - далее будут задания с их применением, что позволит попрактиковаться.
Кстати, чтобы переход по меткам был плавный, добавляйте в CSS это правило:

```
html {
  scroll-behavior: smooth;
}
```

language-css

Lozem ipsum dolor sit amet, consectetur adipisicing elit. Minima quisquam dicta, veritatis ex inventore beatae hic pariatur vitae eius nostrum enim molestias, excepturi perspiciatis exercitationem eum quidem dolores provident facere sapiente corporis. Quae eaque illo debitis maiores quibusdam nihil unde doloremque vitae beatae expedita ducimus, porro voluptates deserunt alias a accusantium nobis? Vitae molestias natus dignissimos? Libero corporis totam tenetur, repudiandae iste consectetur mollitia numquam molestias ex fuga voluptate, blanditiis illo dignissimos ipsa alias cupiditate perferendis dolore. Rerum sunt deleniti aperiam quibusdam iste vitae maiores eveniet aut quo quam optio quaerat deserunt doloribus quos ullam omnis culpa minus delectus dolor voluptatibus nihil, doloremque aliquam esse? Consequuntur repellendus fuga, suscipit voluptatibus facere quis consequatur beatae consectetur ad nulla deserunt tempore eligendi inventore ullam quidem id blanditiis. Nisi omnis quisquam illum esse libero ullam, est, officiis, reiciendis quam minima autem dolor perspiciatis mollitia veniam! Omnis nisi natus doloremque quisquam vel praesentium vitae quaerat, magnam odit, cupiditate et tenetur quas nobis numquam molestias aperiam perferendis maiores fugiat facilis. Quaerat deleniti recusandae, illo suscipit obcaecati totam nesciunt, minus voluptates beatae magni itaque ratione magnam officiis. Asperiores suscipit eaque exercitationem nemo distinctio, quasi voluptatum et officiis hic rem cum. Fugit dicta eaque itaque officiis delectus? Fuga quo fugit ratione, veritatis sit possimus ullam, expedita, incidunt iste amet velit aliquam dolor cumque quisquam eveniet tempora exercitationem odit at labore dolorum! Quo enim veniam corrupti, recusandae eius exercitationem ullam eligendi eveniet perspiciatis

Адресация

Дисклеймер

Тема адресации связана с темой интернет-сетей, что является отдельной дисциплиной в колледжах и университетах, поэтому объяснение ниже будет дано максимально просто.

Пожалуй, одна из сложных тем для новичков, особенно для тех, кто не дружит с компьютером. Адресация нам понадобится не только для ссылок, так что тема важная.

Итак, проще всего в атрибут **href** вставлять URL (Uniform Resource Locator), который является абсолютным адресом документа в сети Интернет. URL был изобретён создателем HTML Тимом Бернерсом-Ли ещё в 1990 году и позволял определять адрес файлов. Сейчас URL определяет адрес всех ресурсов в сети.

Как получить URL?

Всё просто. Если это страница, то мы дважды кликаем по адресной строке в браузере и копируем информацию оттуда. Возьмём для примера адрес-ссылку главной страницы английской Википедии:

https://en.wikipedia.org/wiki/Main_Page

- [https://](#) - это протокол передачи данных;
- [en.wikipedia.org](#) - адрес самого сайта;
- а далее идут каталоги, в которых и находятся статьи. Данный адрес переводит нас на главную страницу.

К примеру, зайдём на статью про дубы, и получим следующую ссылку:

<https://en.wikipedia.org/wiki/Oak>

С английской Википедией всё понятно, но что будет, если мы перейдём на русскоязычную версию? Какой адрес мы получим?

<https://ru.wikipedia.org/wiki/Дуб>

выглядит симпатично. Попробую его скопировать и вставить на Степик:

<https://ru.wikipedia.org/wiki/%D0%94%D1%83%D0%B1> - результат. Почему так?

Потому что символы кириллицы декодируются с помощью Юникода. Если сайт полностью назван кириллицей, к примеру <http://макулатура.рф>, то результат декодирования будет ещё более запутанным - <http://xn--80aaa2bfd6apgd.xn--p1ai/>. Запомним это.

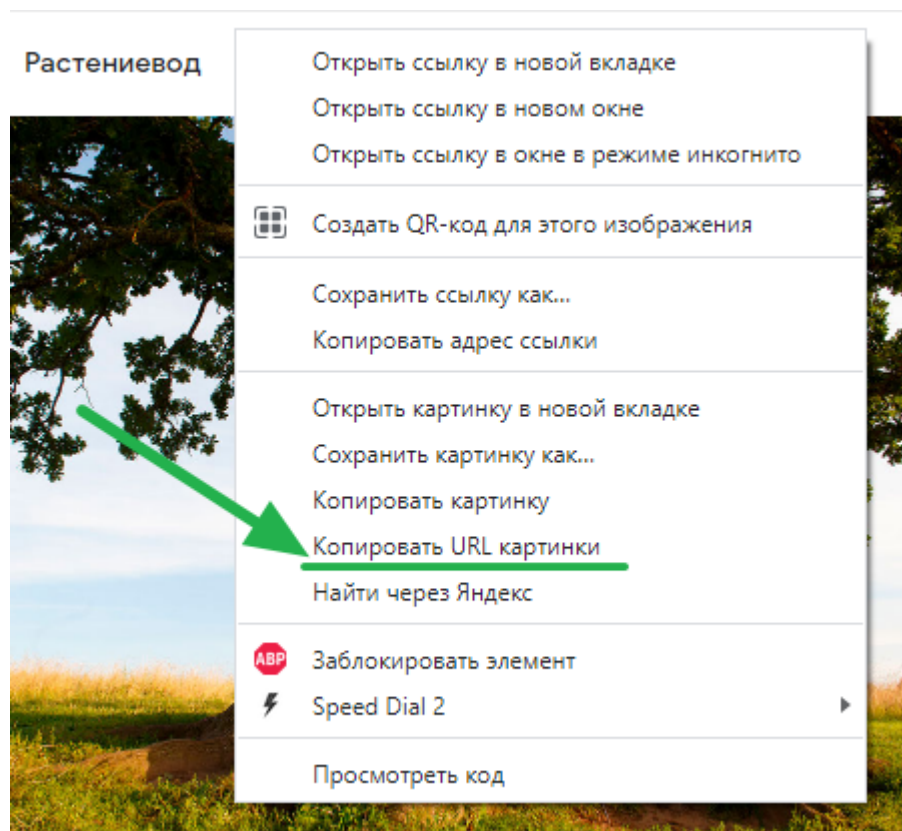
Чтобы всегда получать **полный адрес**, настройте адресную строку своего браузера (в Chrome надо нажать ПКМ по строке и поставить галочку возле "Всегда показывать URL полностью"):

<https://stepik.org/edit-lesson/861617/step/1>

Эмодзи	Win + Точка
Отменить	Ctrl + Z
Вырезать	Ctrl + X
Копировать	Ctrl + C
Вставить	Ctrl + V
Вставить и перейти на страницу https://ru.wikipedia.org/wiki/Дуб	
Удалить	
Выделить все	Ctrl + A
Управление поисковыми системами и поиском по сайту	
✓ Всегда показывать URL полностью	

Как получать адрес картинок?

Если вы пользуетесь поиском изображений в сервисах Яндекса или Гугла, то вам нужно, открыв картинку, нажать по ней ПКМ и скопировать URL изображения:



То же самое с изображениями можно делать и на других сайтах.

Советую также найти бесплатные облачные хранилища, которые могут предоставлять URL-адреса для загруженных туда картинок. К примеру, <https://wampi.ru/>. После загрузки изображения сайт предоставляет коды для встраивания на свою страницу:



Изображение

Для вставки изображения будем использовать одиночный элемент ``.

```
<img src="" alt="">
```

Атрибуты

Как и в случае с ссылкой, у него есть обязательный атрибут, который указывает путь к файлу, но он называется иначе, а именно `src` (*source - источник*).

`alt` - *альтернативный текст*. Он будет выводиться на экран, если не загрузится картинка. Многие не понимают, зачем он нужен, но для современной вёрстки он необходим, потому что предназначен для людей с ограничениями по зрению, которые пользуются браузерами с голосовой

озвучкой контента.

Считается обязательным.

Также с помощью атрибутов `height` и `width` можно менять отображение картинок, делая их меньше или больше (сам размер изображения не меняется). Базово в качестве единиц измерения используются пиксели (px). Если хотите просто уменьшить/увеличить изображение, чтобы оно при этом не растянулось/вытянулось, то изменяйте значение только одного атрибута, а второй изменится пропорционально:

```
`, который группирует внутренние элементы, а также тег `<figcaption>`, являющийся подписью.

```
<figure>

 <figcaption>рис. 1</figcaption>
</figure>
```



рис. 1

Данные теги появились только в HTML5 и, по сути, являются **семантическими**.

\_Браузеры Firefox и Chrome автоматически добавляют к `<figure>` свойство `margin` со значением 1em 40px (1em равен размеру шрифта родительского элемента (по умолчанию - 16px)).

Также как-то идентифицировать изображение можно с помощью атрибута `title`, значение которого будет выводиться при наведении курсором мыши на картинку (так называемая подсказка).

```
<figure>

 <figcaption>рис. 1</figcaption>
</figure>
```

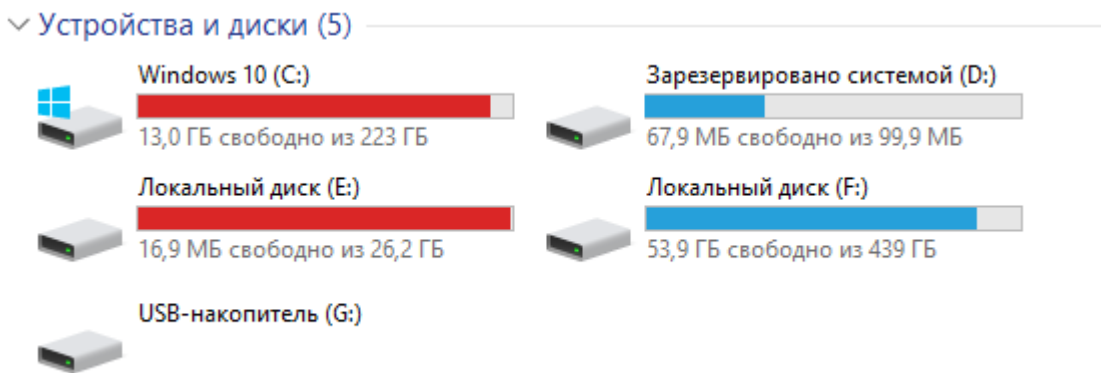


рис. 1 (курсор не заскринился, я его нарисовал)

## Локальные файлы и пути к ним

Что делать, если файл, который нужно открыть или загрузить на сайт, находится на вашем компьютере? В редакторе CodePen ссылаться на такие документы нельзя. В таком случае, мы создаём проект на своём компьютере.

В шаге №2 мы говорили про адресацию в глобальной компьютерной сети. На вашем компьютере (или телефоне) файлы также оформлены в качестве каталогов и имеют определённую адресацию.

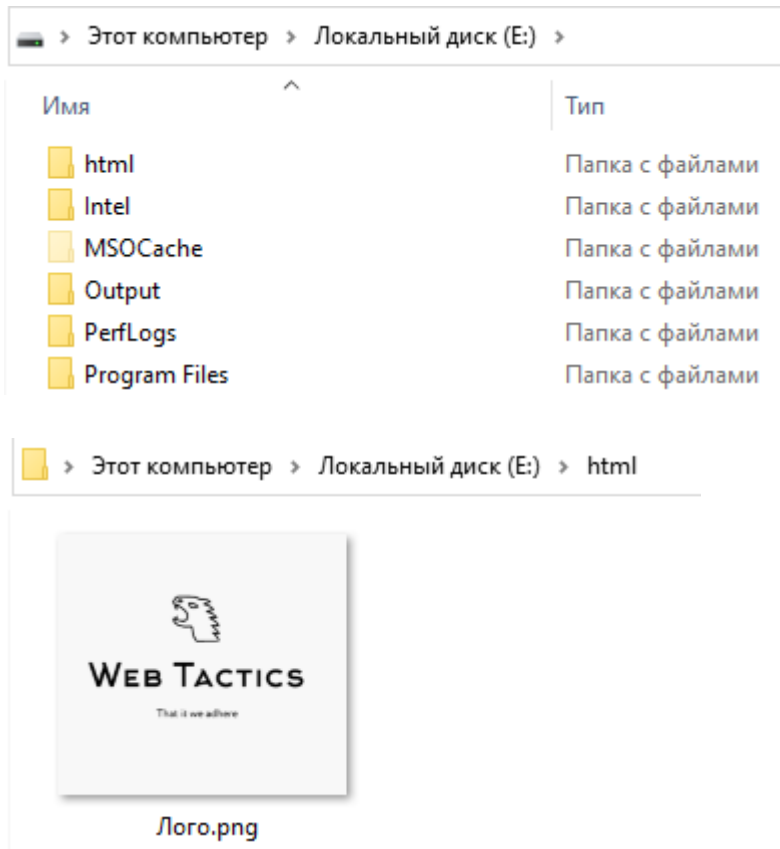


Обычно, программой, которая позволяет увидеть каталоги, является (на Windows) "Этот компьютер" или просто "Компьютер". Эта программа позволяет увидеть физические носители информации. Стандартно, диск, на котором установлена операционная система, называется **системным** и ему присвоена латинская буква "С". Также могут быть и другие физические или виртуальные диски (сюда входят и пути к съёмным USB-накопителям). С них будет начинаться первый вид локальной адресации:

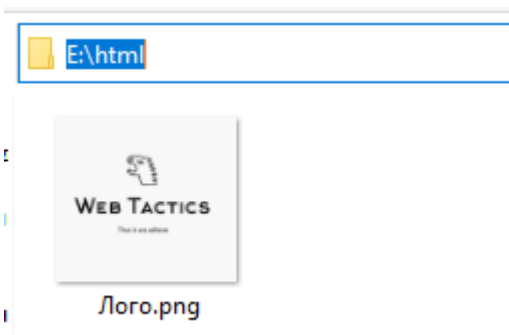
## Абсолютная адресация

У каждого файла на компьютере есть адрес (или путь к нему). Чем дальше (глубже?) файл находится от корневого диска, тем длиннее адрес. Для

примера, зайдём в диск "E" на моём компьютере, а потом в папку html, в которой лежит одна картинка в формате PNG.



В верхнем окошке мы видим изменение пути к файлу. в данном случае оно записывается по-человечески, но чтобы нам получить адрес, который будет понимать компьютер, нужно дважды щёлкнуть по данному окошку и скопировать высветившийся адрес, либо нажать правую кнопку мыши и нажать "Копировать адрес".



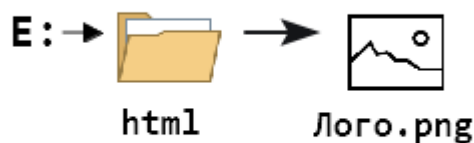
Мы получили короткий адрес, ведущий нас в папку. Обратите внимание на синтаксис - название каталога вначале с двоеточием, потом *обратный слэш* (лучше использовать обычный (см. *подробнее внизу*)) и название папки. Символ слэша ("/") обозначает, что мы что-то "открываем", то есть это действие прохода по каталогу.

Но данного адреса недостаточно, потому что он ведёт в папку, а нам нужно

ссылаться на какой-нибудь документ, чтобы браузер его открыл, к примеру, на Лого. Для этого необходимо в адрес "E:\html" добавить действие открытия файла, причём необходимо обязательно указывать **расширение** файла.

"E:\html\Лого.png" - это абсолютный адрес нашей картинки.

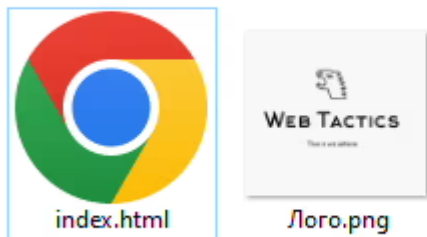
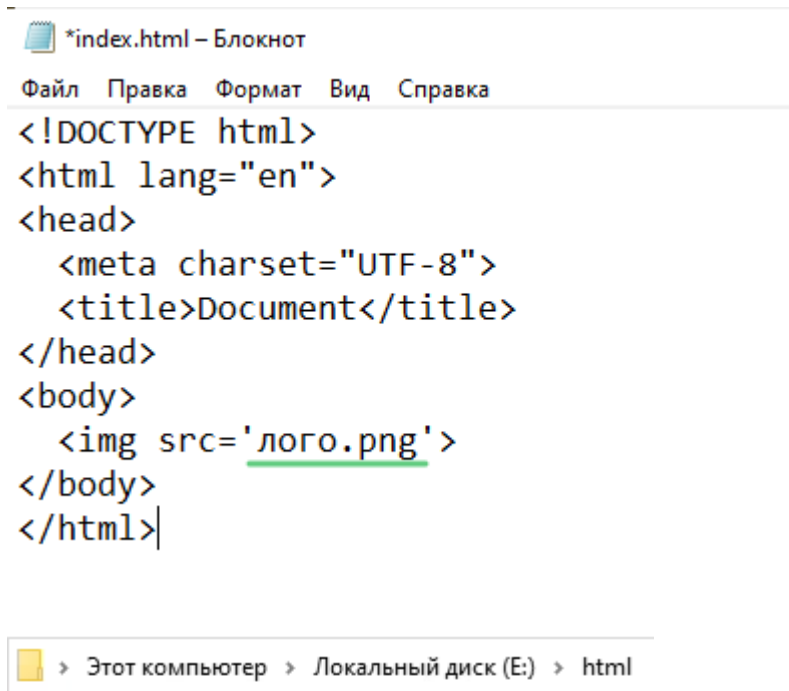
Схематично данный адрес можно изобразить так:



Данную адресацию можно использовать только тогда, когда вы знаете, что ваш проект не будет перемещаться на другие компьютеры и никогда не будет выложен в интернет, иначе ссылки придётся менять. Поэтому, **абсолютную адресацию мы не используем**.

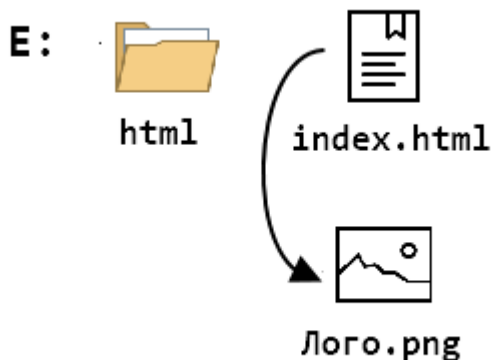
## Относительная адресация

При относительной адресации путь к искомому файлу указывается относительно исходного файла, который у нас открыт. Чтобы исследовать принципы относительной адресации, нам нужно создать html-файл на своём компьютере. Я это сделаю в папке html, вставлю туда базовую структуру (скопирую из CodePen) и вставлю изображение, используя относительную адресацию:



Данная запись адреса сработала, потому что изображение находится в той же папке, что и исходный файл. Если же я перемещу изображение или исходный файл, то адрес будет неверным.

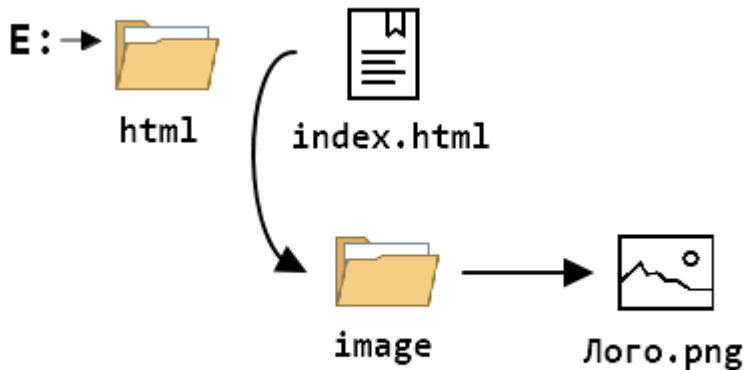
Схематично расположение файлов, папок и путь в данном примере я изобразил так:



Стрелка указывает, с какого файла и на какой мы ссылаемся.

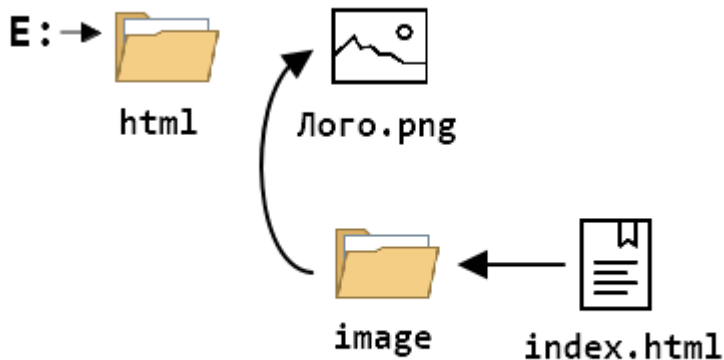
## Вариации

Есть несколько вариантов того, где может находиться искомый файл относительно исходного. Вариантов всего 2 - либо выше, либо ниже:



`image/Лого.png` - на папку "ниже".

В данном варианте я создал папку 'image' и в неё поместил картинку.



`../Лого.png` - на папку выше.

в данном варианте я поменял местами файлы `index.html` и `Лого.png`. Заметьте - чтобы вернуться по каталогу назад (выше), надо указать две точки и слэш.

Более сложные вариации, когда файл находится выше, но в другой папке, пишутся аналогично, используя слэши и парные точки. Такие варианты будут в тестовом задании.

## Про слэш

Есть обычный слэш (*косая черта*, англ. *slash*) и обратный слэш (*обратная косая черта*, англ. *backslash*):

- `"/"` - слэш. Код юникода - `U+002F`. Используется для указания путей к файлу в ОС семейства UNIX, а также в путях http и других веб-протоколах. Не может быть использован для этого в Windows, потому что является служебным символом командной строки;
- `"\"` - бэкслэш. Код юникода - `**U+005C**`. Используется для указания путей к файлу в ОС Windows, MS-DOS, для экранирования символов, в управляющих последовательностях.